Simplifying SAP on i5/OS with Single Sign-on

This document can be found on the web, www.ibm.com/support/techdocs

Version Date: May 6, 2008

IBM Systems & Technology Group

Kolby Hoelzle
hoelzle@us.ibm.com

This page left blank intentionally.

Table of Contents

Preface

Reducing and then keeping administrative costs down is a challenge for most if not all organizations. One common source of high administrative costs is username and password administration. Forgotten passwords and locked usernames not only result in lost productivity, but also a high volume of calls to the help desk. One strategy for reducing username and password administrative costs is single sign-on (SSO). Reducing the number of passwords a user must remember without sacrificing security, is one way that single sign-on can reduce administrative costs. This paper explores the potential for expanding single sign-on to include multiple SAP NetWeaver ABAP environments running on the IBM i5/OS operating system.

About the Author

Kolby Hoelzle is a member of the SAP on i5/OS development team, which is part of the i5/OS development lab in Rochester, Minnesota. He joined IBM in 1999 and has over six years of experience with SAP on the i5/OS platform, including two years working at SAP development in Walldorf, Germany as a member of the joint IBM SAP i5/OS porting team.

Acknowledgements

Thank you to the following reviewers:

Thomas Barlen, IBM STG Lab Services Europe – i5/OS Security and Networking
Michael Frost, IBM STG Lab Services – SAP on i5/OS Practice Leader
Barbara Roth, IBM Server & Technology Group – SAP on i5/OS Porting Team

Introduction

Reducing and then keeping administrative costs down is a challenge for most if not all organizations. One common source of high administrative costs is username and password administration. Forgotten passwords and locked usernames not only result in lost productivity, but also a high volume of calls to the help desk. One strategy for reducing username and password administrative costs is single sign-on (SSO). Reducing the number of passwords a user must remember without sacrificing security through single sign-on is one way to reduce administrative costs.

In a previous paper I discussed expanding single sign-on (SSO) from an SAP® landscape to the entire enterprise utilizing SAP NetWeaver® Enterprise Portal[1]. The portal acted as a single point of access for the entire SAP landscape. This means that a user will access all SAP systems and applications through a single portal interface. Since the portal is its own SAP system, authentication must take place between portal and the other SAP systems in the landscape. This could be accomplished by implementing SAP proprietary single sign-on technology between SAP systems. Expanding single sign-on to the enterprise is then done by enabling single sign-on between the portal and the rest of the enterprise through industry standard protocols such as Kerberos.

This paper explores another alternative. Traditionally, SAP applications have been accessed through an SAP provided thick client named SAP GUI. Each SAP system in the landscape would have its own system logon entry configured. With this scenario there is no single point of access to the SAP landscape, but a separate point of access for each SAP system in the landscape. Even with the release of SAP NetWeaver Enterprise Portals many SAP customers continue to use SAP GUI as the main point of access to the SAP systems in their landscape. So when there is no single point of access to your SAP landscape it becomes beneficial to instead configure a single point of authentication for your SAP landscape. This can be accomplished with single sign-on. This paper explores the benefits and the enablement of single sign-on for SAP NetWeaver AS ABAP™ applications servers using SAP GUI as the client.

Even though this paper discusses some details of implementing single sign-on, it is important that you work with a service provider or someone with single sign-on experience to plan and implement your single sign-on solution. This paper is not meant to be a definitive guide for implementing single sign-on.

Exploring Single Sign-on

When you hear the term single sign-on or SSO, you might first think of eliminating password prompting for end users. Though eliminating these prompts might be popular with the end user, this is not the only reason to implement single sign-on. Three distinct areas are impacted with single sign-on:

---

[1] Kolby Hoelzle, "Expanding Single Sign-on for SAP Landscapes on i5/OS," 23 Oct 2007, <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101127/>.

1. User experience and productivity
2. Administration costs
3. Security

User experience and productivity

This is the most obvious area where single sign-on has an impact. Though sometimes annoying, user and password prompting actually has little impact on productivity. The amount of time a user spends authenticating is a matter of seconds, but if prompting is excessive then the user experience can be negatively impacted. The real impact to productivity is when a user must manage multiple usernames and passwords. A typical user may have up to ten or more usernames and passwords for typical business needs. Combine the number of different passwords with security policies that require passwords be changed on a regular basis and keeping track of passwords can become a big task. Not only can this negatively impact the user's experience, but it has the potential for impacting productivity. More passwords to remember will increase the chance of forgotten passwords, leading to even more impacts on productivity. Single sign-on simplifies the management of passwords for end users and can reduce password related issues that might impact productivity.

Administration costs

Perhaps the biggest impetus for implementing single sign-on is the potential savings in the overall administration costs to an organization. For many organizations the highest percentage of calls to the help desk are for password-related issues. In these situations, not only is the user's productivity impacted, but additional costs are incurred when the help desk or the system administrator must get involved. More than providing end users with a better user experience, single sign-on is an administration strategy that can reduce the number of password related problems and reduce costs. This is accomplished by simplifying the management of usernames and passwords for both the user and the administrators.

Security

Improving security might be the least obvious reason for single sign-on, but it could be the most important. The more usernames and passwords that a user must track, the higher the likelihood a user might intentionally or unintentionally compromise security. For example, a user might be tempted to write down or cache an unsecured list of usernames and passwords or even set trivial, easy to guess passwords. The temptation might even increase if an organization has implemented a security policy that forces users to change passwords on a regular basis. Single sign-on will improve security by reducing the number of passwords and potentially the number of usernames a person must remember.

Authentication versus Authorization

Before any in-depth discussion on single sign-on it is important to understand the difference between authentication and authorization. Simply put, authentication is proving you are who you say you are and authorization is the right to access certain system resources after authentication. A user authenticates by providing credentials, in

most cases a valid username and password. Other methods of authenticating include biometrics, ID cards, and security tokens.

Authorization, on the other hand, is normally transparent to the user and is usually only noticed when access to a specific resource is denied. It is entirely possible for a user to successfully authenticate, but not have authorization to any resources on a system.

Traditionally data used to validate a user's authentication and the data used to determine a user's authorization are stored in the same database or registry. Host systems, networks, and sometimes even applications will have their own user registries specifically for authentication and authorization. In some single sign-on configurations a key design point is to separate the data for authentication and the data for authorization into two separate and distinct repositories. This separation of authentication data and authorization data allows multiple authentication repositories to be merged creating a centralized location to manage all of the users in an enterprise. Authentication data may even be located somewhere physically separate from the resource being accessed.

In contrast to data used for authentication, the data used for authorization will almost always remain with the resource and will remain separate from other user authorization repositories. This is necessary due to differences in authorities and how they are managed from one resource to the next. This also provides additional flexibility since a user may have different levels of authority from one resource to another. Since a user's authorities are rarely changed after being established, having multiple authorization repositories has little impact on maintenance costs.

Password Elimination

User experience and productivity, administration costs, and security are addressed by single sign-on through the elimination of passwords. Rather than attempt to synchronize multiple passwords, eliminating passwords altogether will greatly simplify username and password management for the end user and the administrators. Single sign-on provides an authentication mechanism that allows a user to authenticate once and obtain the necessary credentials allowing automatic authentication to other resources throughout the enterprise, all without sacrificing security. If resources on the network trust and support the authentication mechanism then the passwords tied specifically to these resources can be eliminated in exchange for the authentication mechanism. Depending on the application, this could actually mean disabling the password function altogether and only allowing a user to authenticate through the designated single sign-on mechanism.

A single sign-on solution might even provide a username mapping service, such as IBM® Enterprise Identity Mapping (EIM). These services map one username to another in cases where the usernames for the same user are not identical. This further simplifies administration by allowing an administrator to manage people instead of usernames.

Regardless of how single sign-on is implemented it is important to stay focused on simplifying usernames and password management for the administrators and the end users. This is accomplished through the elimination of passwords without sacrificing

security. As the number of passwords a user must remember decreases, the potential cost savings to an organization increases.

Technical Overview

Enabling single sign-on in an SAP ABAP environment requires the use of multiple different components all working together. These components are interdependent and must be configured correctly for single sign-on to be successful in this environment. In addition to components for both the clients and the servers, the SAP landscape must be configured to participate in a single sign-on landscape that includes non-SAP systems. In this section I provide an overview of the basic components of my single sign-on solution.

The Generic Security Service API and Kerberos

The Generic Security Service API or GSS API is, as its name suggests, a standardized security API. Like any API, the interface is standard in order to enable increased portability, but the underlying implementation will vary from vendor to vendor. The GSS API allows application developers to integrate security into their application using one standard interface instead of a different interface for each platform. The underlying authentication mechanism used for the implementation of the GSS API is left to the operating system platforms or to vendors that specialize in security. IBM's implementation of the GSS API uses the Kerberos protocol for the authentication mechanism.

Kerberos is a distributed authentication protocol that provides the foundation for the solution presented in this paper. Kerberos was originally developed at Massachusetts Institute of Technology (MIT) and is widely used in many commercial applications including versions of Microsoft® Windows® and IBM i5/OS®. The protocol is named after the mythical three headed dog Cerberus, who was the guardian of the gates to the underworld in Greek mythology[2].

The Kerberos protocol consists of three distinct parts: authentication server, ticket granting server, and services. The authentication server and ticket granting server, though logically distinct, often exist on the same physical machine and are collectively referred to as the key distribution center or KDC. The authentication server verifies user credentials against a database or directory service (note that in Kerberos terminology a user is known as a principal). In this case the credentials are a username and password. If the authentication server can successfully authenticate a user, then the ticket granting server grants a special ticket to the user called a ticket granting ticket or initial ticket. The lifetime of this ticket is temporary and will expire after a configured period of time, usually about eight to ten hours.

Services are Kerberos enabled resources such as applications or host systems that a user will need to access. A Kerberos enabled resource is one that has the capability to accept Kerberos tickets. Once a user is granted a ticket granting ticket, that user can now attempt to authenticate with a Kerberos enabled service by requesting a service ticket.

---

[2] Garman, Jason. *Kerberos: The Definitive Guide*. 1st edition, Sebastopol, CA: Farnham: O'Reilly 2003.

The ticket granting ticket and the service ticket request are both sent to the KDC, which will then grant a service ticket. If the service can successfully validate the service ticket, then the user will have authenticated with that service. This is all done without sending a username or password over the network. Like ticket granting tickets, service tickets are temporary and will expire after a configured period of time. This forces the user to re-authenticate and limits the opportunity for tickets to be intercepted and used by someone other than the intended user.

Keytab

A small, but essential component of Kerberos is what is known as the keytab. In the Kerberos protocol, users and resources must be identified by unique names (principals). This includes services as well as users. This is business-as-usual for users since they already have associated usernames and authentication mechanism in place – normally a password challenge. For services this concept is somewhat foreign, since most services don't have an associated username and password through which to authenticate. In a Kerberos realm a participating resource is given a unique identity known as the service principal name (SPN). The SPN also has an associated password. Since an SPN normally will not have an associated person to enter a password, its password is stored in an encrypted file called the keytab. The keytab is created as part of the Kerberos client configuration, but it must be maintained when a new SPN is created for a particular host.

Secure Network Communications

Secure Network Communications (SNC) is an SAP security framework that enables the integration of external security products with SAP systems[3]. External security products may be provided by the operating system or by third parties that specialize in security. Integrating external security products can strengthen security by providing security functions not directly available with SAP systems.

The SNC framework is based on the GSS API. This standard interface provides the flexibility to plug-in and leverage different implementations of the GSS API. As long as SNC recognizes the underlying authentication mechanism the GSS API implementation can be used directly with the SNC. Implementations of the GSS API using authentication mechanisms that are not recognized can still be utilized, but a specialized SNC adapter will need to be used. SNC recognizes Kerberos, so an SNC adapter is not necessary when using the IBM implementation of the GSS API. More information about SNC and the SNC adapter can be found on the SAP Developer Network under Network Security[4].

Single Sign-on Test Environment

To demonstrate single sign-on, I created a simple test environment. My environment consists of three basic components: SAP NetWeaver landscape running on i5/OS; Microsoft Windows network and workstations; and i5/OS systems not running SAP. For

[3] SAP AG. SAP Help Portal: Secure Network Communications (SNC). <http://help.sap.com/saphelp_nw04s/helpdata/en/e6/56f466e99a11d1a5b00000e835363f/content.htm/>
[4] SAP AG. SAP Developer Network: Technology – Network Security (BC-SNC). <https://www.sdn.sap.com/irj/sdn/sdnservices/icc?rid=/webcontent/uuid/e112cb72-0501-0010-63a3-f45326c176ae/>

my SAP landscape there is no single point of access to the landscape and each SAP system maintains its own user repository. SAP's Central User Administration (CUA) was not configured for my environment, but CUA and single sign-on are compatible and complimentary technologies that can be configured together.

My SAP landscape consists of two SAP NetWeaver 7.0 systems with one running on i5/OS V5R3 and the other running on i5/OS V5R4. Both NetWeaver systems are Add-in systems running both AS ABAP and AS Java. Windows workstations are the primary access point to all resources. All of the Windows workstations are part of a Windows domain. The Windows domain controller is running on Windows Server® 2003 Enterprise Edition and the workstations are a combination of Windows Server 2003 and Windows XP Professional. The additional i5/OS system is not running any SAP applications and is running on i5/OS V6R1. This system represents legacy applications.

In this environment our user John arrives at work and signs on to his personal workstation with his Windows domain username, jsmith, and his password. After checking his e-mail John needs to do some work on one of the SAP systems. John starts SAP Logon and selects an SAP system. SAP GUI starts and presents a dialog asking for a username and password. John knows that for this particular SAP system, his SAP username is different than his Window username. John types in the username johns and his password and then signs on to the system. A short time later, John remembers that he has some work to do on the legacy system. Though John can use iSeries Navigator to do his work, he likes to work directly on the i5/OS. John opens a 5250 telnet window and enters his username smithj which happens to be different than both his Windows username and the SAP username he just used. Later that afternoon John needs to access a different SAP system. This time the SAP username matches his Windows username. John types in his username, but remembers that he changed the password the previous day. Even though John tries to keep his passwords in synch, sometimes this is hard to do. John knows this password is the same as one of his other passwords, but he can't quite remember which one. After two unsuccessful tries, he finally gets it right. He signs on and saves himself a call to the help desk, see figure 1.

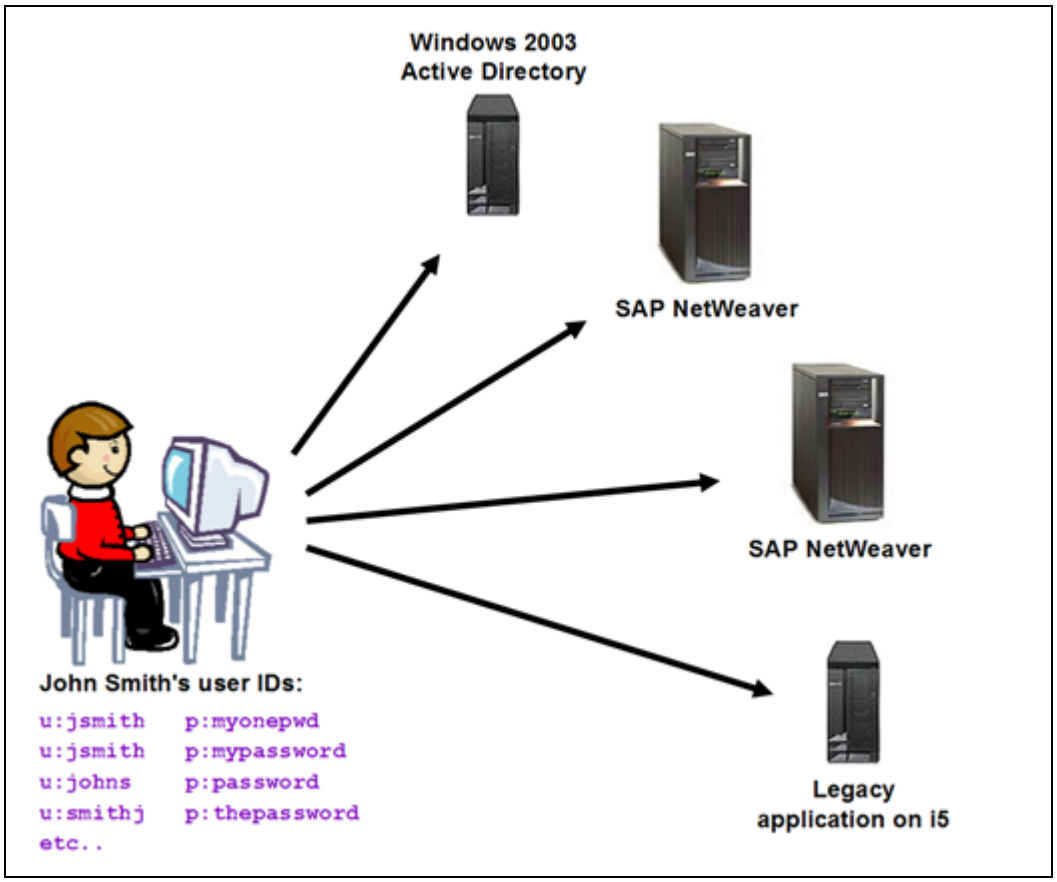

Figure 1. Enterprise environment with multiple points of access

In John's company each time a new type of system was brought online, a different convention was used when creating usernames, resulting in slightly different usernames for many of the users. John finds himself getting frustrated occasionally when trying to keep track of his many usernames and passwords. It got so bad at one point that John started putting his passwords under his keyboard, but he decided that wasn't a good idea when an e-mail went around warning employees not to do this because of the potential security breach. This is when John decided to look into single sign-on.

Implementation

This section discusses an implementation overview for setting up a single sign-on environment for an SAP on i5/OS landscape. Even though this paper discusses some details of implementing single sign-on, it is important that you work with a service provider or someone with single sign-on experience to plan and implement your single sign-on solution. This paper is not meant to be a definitive guide for implementing single sign-on.

There are multiple components that must be configured for a single sign-on solution. It is entirely possible that some resources in your enterprise might not be configured with single sign-on while other parts will. The feasibility of enterprise wide single sign-on depends to a large degree on the resources found in your enterprise and whether or not

they support single sign-on protocols such as Kerberos. Since this paper focuses on SAP ABAP applications servers running on i5/OS, this implementation section only covers configuring components essential to this environment.

A key requirement to the single sign-on solution described in this paper is the utilization of a Windows domain based on Windows 2000 or later releases (Windows 2000 was the first release with integrated Kerberos support). Any Windows workstations used for single sign-on must be running Windows 2000 or later releases and be a member of a Windows domain. The requirement to be a member of a Windows domain is based on the assumption that the initial point of access to an enterprise's network resources will be a Windows workstation. This means that a user will always sign-on to a Windows workstation before accessing any other resources.

Another requirement is i5/OS support for the GSS API. This support is provided with the i5/OS licensed product 5722-NAE "Network Authentication Enablement" on V5R4 and later releases. For V5R3 the support is provided with the i5/OS licensed product 5722-AC3 "Crypto Access Provider 128-bit".

Network Authentication Services

Network Authentication Service (NAS) provides the i5/OS implementation of Kerberos and the GSS API. NAS is an integrated part of i5/OS, but it must be configured. NAS allows i5/OS users the ability to use Kerberos to access the i5/OS through a telnet client or Kerberos-enabled client-server applications running on i5/OS, such as iSeries® Navigator. NAS configuration must be done on each i5/OS system that will participate in single sign-on. Configuration is done through an iSeries Navigator wizard. To start the wizard, do the following:

1. Start iSeries Navigator
2. Expand the i5/OS system where you will configure NAS
3. Expand Security
4. Right click Network Authentication Service
5. Select Configure to start the wizard

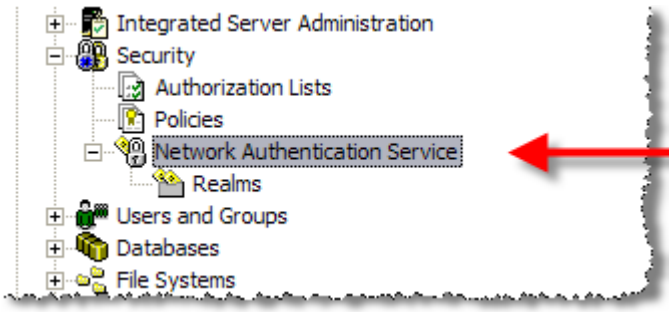

Figure 2. Starting the NAS wizard

More information on configuring NAS can be found in the i5/OS Information Center[5].

Since SAP NetWeaver runs in the i5/OS PASE[6] environment, a Kerberos client running in the PASE environment is also necessary. This client is not configured as part of the NAS configuration and must be configured manually. Fortunately, there are some i5/OS provided scripts that simplify configuration.

The PASE based Kerberos client must be configured on each i5/OS system that is hosting an SAP application server. To configure the PASE-based Kerberos client, run the following commands from the i5/OS command line.

1. Start the PASE shell interpreter
   ```
   CALL QP2TERM
   ```
2. Setup the Kerberos environment
   ```
   export PATH=$PATH:/usr/krb5/sbin
   ```
3. Run the Kerberos configuration script config.krb5
   ```
   config.krb5 -C -r <kerberos_realm> -d <domain> -c
   <key_distribution_center> -s <server>
   ```

   - `<kerberos_realm>` = name of the Kerberos realm, usually the same as the Windows domain, except all upper case. This will be the same as the Kerberos realm configured for NAS.
   - `<domain>` = fully qualified Windows domain name
   - `<key_distribution_center>` = in a Windows domain this will be the fully qualified name of the domain controller
   - `<server>` = name of the Kerberos admin server, this will be the primary domain controller and will usually have the same value as the `<key_distribution_center>`

Figure 3 shows the steps for setting up a Kerberos client in the realm DEPTGGQ.RCHLAND.IBM.COM and using a Windows server named erppcx20.rchland.ibm.com as the KDC. Note that the Windows domain and the DNS domain do not have to be the same.

---

[5] International Business Machines Corporation. IBM System i and i5/OS Information Center: Network Authentication Service. <http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzakh/rzakh000.htm?resultof=%22%/>.

[6] International Business Machines Corporation. IBM System i and i5/OS Information Center: i5/OS PASE. <http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzalf/rzalfintro.htm?resultof=%22%50%41%53%45%22%20/>

```
  $
> export PATH=$PATH:/usr/krb5/sbin
  $
> config.krb5 -C -r DEPTGGQ.RCHLAND.IBM.COM -d deptggq.rchland.ibm.com -c erppc
  x20.rchland.ibm.com -s erppcx20.rchland.ibm.com
  Initializing configuration...
  Creating /etc/krb5/krb5_cfg_type...
  Creating /etc/krb5/krb5.conf...
  The command completed successfully.
  $
```

Figure 3. Configuring a PASE Kerberos client

In order to simplify the management of NAS and the PASE Kerberos client, configure the PASE Kerberos client to use the same keytab as the NAS Kerberos client.

1. Create a symbolic link from the PASE client keytab pointing to the NAS keytab. Run the following command from the i5/OS command line.

```
ADDLNK
OBJ('/QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab')
NEWLNK('/etc/krb5/krb5.keytab')
```

2. Change the owner of the new link to QSYS

```
CHGOWN OBJ('/etc/krb5/krb5.keytab') NEWOWN(QSYS) SYMLNK(*YES)
```

Your PASE Kerberos client should now be configured.

Windows Domain Controller and Active Directory

Microsoft Windows Active Directory® is an integrated part of Windows Server technology and is a key component of the Windows Server domain controller. A domain controller is required for any Windows based network. Active Directory is used as the user repository for both authentication and authorization for Windows-based resources in a Windows network. Active Directory is a critical part of a Windows network making it an essential part of this single sign-on solution.

The Windows domain controller also acts as a Kerberos Key Distribution Center (KDC). All three components of the KDC: authorization server, ticket granting server, and Kerberos database will run on the domain controller. In this case Active directory acts as the Kerberos database.

Since Kerberos is integrated into Windows operating systems, it is not necessary to configure the Windows domain controller to use Kerberos. However, it is necessary to create Active Directory accounts for the servers hosting SAP applications. These Active Directory accounts must then be mapped to a Kerberos service principal name (SPN). An SPN is used by Kerberos to identify a resource on the network.

First create a new computer account for each i5/OS systems hosting SAP application servers using the Active Directory GUI, see figure 4.

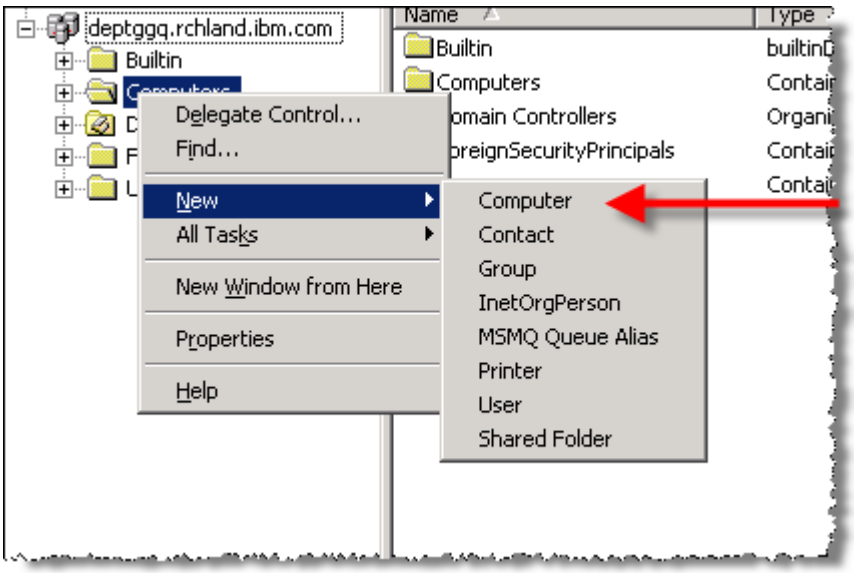

Figure 4. Creating a computer account in Active Directory

The command line tool DSADD can also be used to create Active Directory accounts. This is especially helpful, for automation using batch files. More information on administering accounts in a Windows domain can be found on Microsoft TechNet[7].

Once the Active Directory account is created, then it is time to map the account to the SPN. This is done through a command line tool called KTPASS. There is no GUI interface to this tool, so it must be run from a DOS prompt on the domain controller or via a batch file. Before running KTPASS download the latest version of the Windows Support tools at http://www.microsoft.com/downloads/details.aspx?familyid=96A35011-FD83-419D-939B-9A772EA2DF90&displaylang=en

The KTPASS command has the following syntax:

```
KTPASS -MAPUSER <host>$@<windows_domain> -PRINC
<service>/<host>.<fqdn>@<kerberos_realm> -PASS <my_password> -mapop set
-ptype KRB5_NT_SRV_HST +answer -out <dir>\<host>_SAPService.keytab
```

- `<service>` = arbitrary name identifying the service, I recommend SAPService
- `<host>` = DNS name of server hosting the SAP systems
- `<windows_domain>` = fully qualified Windows domain name
- `<fqdn>` = fully qualified DNS domain name (this might be different than the Windows domain
- `<kerberos_realm>` = the name of the Kerberos realm
- `<my_password>` = arbitrary password
- `<dir>` = location of the generated output file

[7] Microsoft Corporation. Microsoft TechNet: Networking and Access Technologies. <http://technet.microsoft.com/en-us/network/default.aspx/>.

- The `-MAPUSER` parameter specifies the Active Directory account, the dollar sign ($) after the host specifies that this is a computer account
- The `-PRINC` parameter specifies the SPN.
- The `-PASS` parameter specifies the password used to generate the encryption key
- The `mapop` parameter needs to be set to "set" for the initial mapping to an Active Directory account. For subsequent mapping to the same Active Directory account mapop should be set to "add".
- The `+DesOnly` parameter specified that only DES encryption should be used
- The `-crypto` parameter specifies the encryption type
- The `-ptype` parameter specifies the Windows domain account type
- The `+answer` parameters specifies to automatically answer prompts
- The `-out` parameter specifies the location of a the generated keytab entry

Figure 5 shows the command used to create a computer account for a computer named ERPH2 using DSADD. The SPN SAPService/erph2.rchland.ibm.com is then mapped to the computer account ERPH2 using KTPASS. The KTPASS command produces an output file, specified by the `-out` parameter. This output file contains a keytab entry and will be used to complete the configuration on the server hosting the SAP application.

```
C:\>DSADD computer cn=ERPH2,cn=computers,dc=DEPTGGQ,dc=RCHLAND,dc=IBM,dc=COM
dsadd succeeded:cn=ERPH2,cn=computers,dc=DEPTGGQ,dc=RCHLAND,dc=IBM,dc=COM

C:\>KTPASS -MAPUSER erph2$@deptggq.rchland.ibm.com -PRINC SAPService/erph2.rchla
nd.ibm.com@DEPTGGQ.RCHLAND.IBM.COM -PASS singleso -mapop set -ptype KRB5_NT_SRV_
HST +answer -out \temp\erph2_SAPService.keytab
Targeting domain controller: erppcx20.deptggq.rchland.ibm.com
Using legacy password setting method
WARNING: Account ERPH2$ is not a normal user account (uacFlags=0x1020).

Do you really want to delete any previous servicePrincipalName values on ERPH2$
[y/n]?  auto: YES
Successfully mapped SAPService/erph2.rchland.ibm.com to ERPH2$.
WARNING: Resetting ERPH2$'s password may cause authentication problems if ERPH2$
 is being used as a server.

Reset ERPH2$'s password [y/n]?  auto: YES
Key created.
Output keytab to \temp\erph2_SAPService.keytab:
Keytab version: 0x502
keysize 91 SAPService/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM ptype 3 (KRB
5_NT_SRV_HST) vno 2 etype 0x17 (RC4-HMAC) keylength 16 (0xae059df46152e01ac62248
698e2a1eeb)

C:\>
```

Figure 5. Mapping a computer account to a Kerberos service principal name

Adding a New Entry to the Keytab

The newly created keytab entry must be merged with the existing keytab on the i5/OS server hosting the SAP applications. To do this, first copy the generated keytab from the domain controller to the host system. Once the keytab file is on the host use the ktutil utility to merge the keytab.

1. Start the PASE shell interpreter
   ```
   CALL QP2TERM
   ```
2. Setup the Kerberos environment
   ```
   export PATH=$PATH:/usr/krb5/sbin
   ```

3. Start the ktutil utility
```
ktutil
```
4. Read the new keytab entry into the ktutil buffer
```
rkt <new_keytab>.keytab
```
5. Write the contents of the ktutil buffer out to the existing keytab
```
wkt /etc/krb5/krb5.keytab
```
6. Clear the ktutil buffer and then verify that the new entry was added to the keytab
```
clear
rkt /etc/krb5/krb5.keytab
list
```

Figure 6 shows the steps to merge a keytab entry into an existing keytab using the ktutil utility. Figure 7 shows the newly added entry in the keytab.

```
  $
> export PATH=$PATH:/usr/krb5/sbin
  $
> ktutil
  ktutil:
> rkt /usr/sap/SSO/erph2_SAPService.keytab
  ktutil:
> wkt /etc/krb5/krb5.keytab
  ktutil:
> clear
  ktutil:
> rkt /etc/krb5/krb5.keytab
  ktutil:
```

Figure 6. Merging a keytab entry with an existing keytab

```
  ktutil:
> list
  slot   KVNO   Principal
  ------ ------ ---------------------------------------------------------
       1      1 krbsvr400/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM
       2      1 krbsvr400/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM
       3      1 krbsvr400/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM
       4      2 SAPService/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM
  ktutil:
```

Figure 7. List the contents of the keytab, including the new entry

For help on the ktutil command type a question mark (?) at the ktutil prompt.

Now that the keytab has been updated and the SPN has been created on the KDC the configuration can be tested. This is done by requesting a Kerberos ticket for the SPN using the keytab. This is done from the PASE environment with the kinit command.

1. Start the PASE shell interpreter
```
CALL QP2TERM
```
2. To use the kinit command it is not necessary to initialize the Kerberos environment. You can simply run kinit
```
kinit -k SAPService/<host>.<fqdn>
```

- `<host>` = the host name
- `<fqdn>` = the fully qualified DNS domain.

The command only produces output if there are errors. So in this case no news is good news. If the command does run successfully, then the KDC has provided credentials for the SPN.

SAP Application Server

With a Kerberos foundation in place, it is now time to configure the SAP application servers to take advantage of Kerberos and enable single sign-on. This configuration must be done for each application server. Configuring the application servers to use Kerberos requires modifications to both the start and instance profiles.

Instance Profiles

Most likely the required profile parameters will not exist and will have to be added to the instance profile. Most of the profile parameters are used to enable SNC, but some are used to adjust the behavior of the SNC. The following parameters enable the SNC and must be added to the instance profile:

```
snc/enable = 1
snc/gssapi_lib = /lib/libgssapi_krb5.a(libgssapi_krb5.a.so)
snc/identity/as = p:<spn>@<kerberos_realm>
```

- `<spn>` = the Kerberos service principal name.
- `<kerberos_realm>` = name of the Kerberos realm

The parameter `snc/enable` turns SNC on or off. The parameter `snc/gssapi_lib` identifies the fully qualified path of the GSS API implementation to be used for SNC. The portion in parenthesis is part of the parameter value and needs to be included. The final parameter, `snc/identity/as` is the Kerberos service principal name that will be used by SNC for authentication.

The instance profile parameter, `snc/accept_insecure_gui`, should also be set. The parameter affects SNC behavior and can be set to '`0`', '`1`', or '`U`'. Don't worry this is not a typo, the wording for the parameter really is insecure as opposed to unsecured. This parameter determines whether single sign-on is absolutely required for single sign-on (parameter to 0); whether single sign-on can be used, but is not necessary (parameter set to 1); or whether some users are explicitly permitted to sign-on with a username and password that is stored on the SAP system while other users not specified must still use single sign-on (parameter set to U). I would recommend setting this parameter to U and allowing certain specialized users such as administrators the ability to sign-on without single sign-on. However, initially this parameter must be set to 1 to complete the configuration. SNC must be enabled and an administrator must be able to sign-on to complete the configuration. If this parameter is set to 0 or U when SNC is enabled, no one will be able to sign-on to complete the configuration. Once the single sign-on configuration is complete the value for `snc/accept_insecure_gui` should be set to '`U`' and the SAP system restarted. Figure 8 shows the profile parameter values for the SPN SAPService/erph2.rchland.ibm.com.

snc/enable	1
snc/gssapi_lib	/lib/libgssapi_krb5.a(libgssapi_krb5.a.so)
snc/identity/as	p:SAPService/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM
snc/accept_insecure_gui	U

Figure 8. Profile parameter settings enabling SNC

Start Profile

Once SNC is enabled, your SAP system will require a Kerberos ticket in order to start. This is very important and worth repeating, SAP applications will not start if a valid Kerberos ticket for the SAP application SPN is not available. This ticket is acquired by running the kinit command for the SPN created for SAP applications. The user requesting the credentials must be the i5/OS user under which the SAP application runs. This user will always be <sid><nn>, where <sid> is the SAP system ID and <nn> is the instance number. This user has limited capabilities and cannot be used to start an interactive session. To request a ticket using <sid><nn>, the following SBMJOB command can be used.

```
SBMJOB CMD(CALL PGM(QP2SHELL) PARM('/QOpenSys/usr/bin/kinit' '-k'
'<spn>@<kerberos_realm>')) JOB(KINIT) USER(<sid><nn>)
```

- `<spn>` = the service principal name
- `<kerberos_realm>` = the name of the Kerberos realm
- `<sid>` = the SAP system ID
- `<nn>` = the instance number

To ensure that valid tickets are available whenever starting SAP, add the SBMJOB command to the start profile. Figure 9 shows an entry in the start profile for an SAP system named S37 with an instance number of 37.

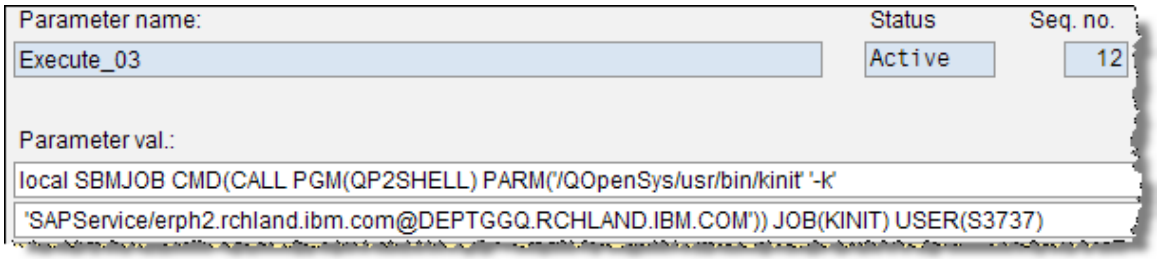

Figure 9. Use the start profile to automatically request Kerberos ticket on startup

For security reasons, Kerberos tickets expire after a certain period of time and will need to be renewed. Expiring tickets significantly decrease the probability that an intercepted ticket can be decrypted and used to gain access to a resource. Since the tickets expire for the SPN, it is necessary to renew these tickets regularly. This can be done manually or automatically, though I recommend renewing tickets automatically. On the i5/OS, scheduled jobs can be used to automatically renew tickets. To create a scheduled job to renew tickets, run the following command from the i5/OS command line:

```
ADDJOBSCDE JOB(KINIT) CMD(CALL PGM(QP2SHELL)
PARM('/QOpenSys/usr/bin/kinit' '-k' '<spn>@<kerberos_realm>'))
```

```
FRQ(*WEEKLY) SCDDATE(*NONE) SCDDAY(*ALL) SCDTIME('00:00:00')
JOBD(R3<sid>400/R3_<nn>) USER(<sid><nn>) TEXT('Renew Kerberos tickets')
```

To ensure that valid tickets are always available, it may be necessary to create multiple scheduled job entries. By default Kerberos tickets expire every ten hours, so I recommend renewing tickets three times per day every eight hours. However, since ticket lifetime is configurable, the renewal interval should be adjusted according to your configuration.

Grant Authority to <sid><nn>

The final step is to grant the necessary authorities to the i5/OS user <sid><nn>. The user profile <sid><nn> must have access to the following Kerberos components: configuration file, keytab, and the credential cache. The i5/OS command CHGAUT can be used to grant the necessary authorities. The following commands should be run for each <sid><nn> user profile:

- `CHGAUT OBJ('/var/krb5/security/creds') USER(<sid><nn>) DTAAUT(*RWX)`
- `CHGAUT OBJ('/QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab') USER(<sid><nn>) DTAAUT(*R)`
- `CHGAUT OBJ('/etc/krb5') USER(<sid><nn>) DTAAUT(*RX)`

Figure 10 shows the results of granting the necessary authority to user profile S3737.

```
> CHGAUT OBJ('/var/krb5/security/creds') USER(S3737) DTAAUT(*RWX)
  Object changed.
> CHGAUT OBJ('/QIBM/UserData/OS400/NetworkAuthentication/keytab/krb5.keytab
  ') USER(S3737) DTAAUT(*R)
  Object changed.
> CHGAUT OBJ('/etc/krb5') USER(S3737) DTAAUT(*RX)
  Object changed.
```

Figure 10. Granting authority to <sid><nn>

Your SAP system is now ready to use SNC. Restart the system to activate SNC.

SAP users

After SNC is enabled, each individual user profile on the SAP system must be configured to utilize SNC and enable single sign-on. This is done from within the SAP application. Sign-on to the SAP application as a user with sufficient authority to modify SAP user profiles. Using transaction SU01, edit each user profile to enable SNC by doing the following.

1. Select the SNC tab. You will see the SNC tab in SU01 only if SNC is enabled.
2. In the SNC name field enter `p:<user>@<kerberos_realm>`
   - `<user>` = a Windows domain user, this does not have to be the same as the SAP user
   - `<kerberos_realm>` = the name of the Kerberos realm
3. Click save

If the SNC name is correct a green checkmark will appear next to the label “Canonical name determined” in the SNC data box.

Optionally, you can select the box “Unsecure communications permitted (user-specific)” to allow this specific user to sign-on without using single sign-on. This should only be allowed for certain users such as administrators. Figure 11 shows the configuration of user KOLBYJH in Kerberos realm DEPTGGQ.RCHLAND.IBM.COM. Notice that KOLBYJH does not have the ability to sign-on without single sign-on.

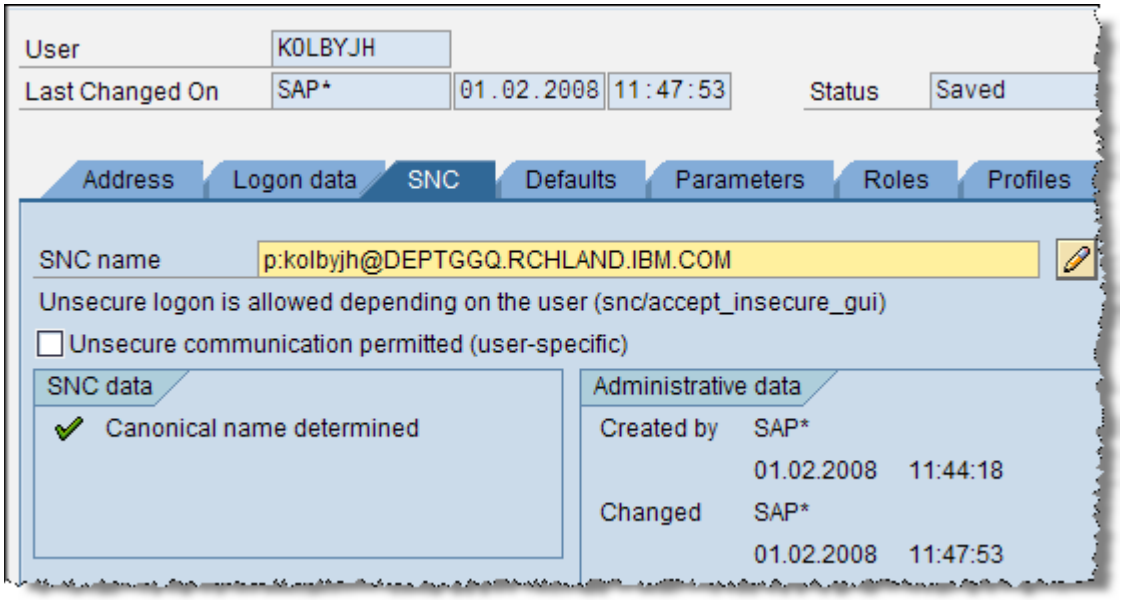

Figure 11. Configuring specific users to use single sign-on

By only allowing users the ability to sign-on with single sign-on you can eliminate the passwords for those users and reduce costs associated with password problems. After single sign-on is enabled and the user profiles are configured, passwords can be deactivated for all users that will only use single sign-on to access the system. To deactivate the passwords do the following.

1. Select the Logon data tab from SU01
2. Click the deactivate button (looks like a flashlight) next to the initial password field
3. Click save.

Figure 12 shows password deactivation for user KOLBYJH.

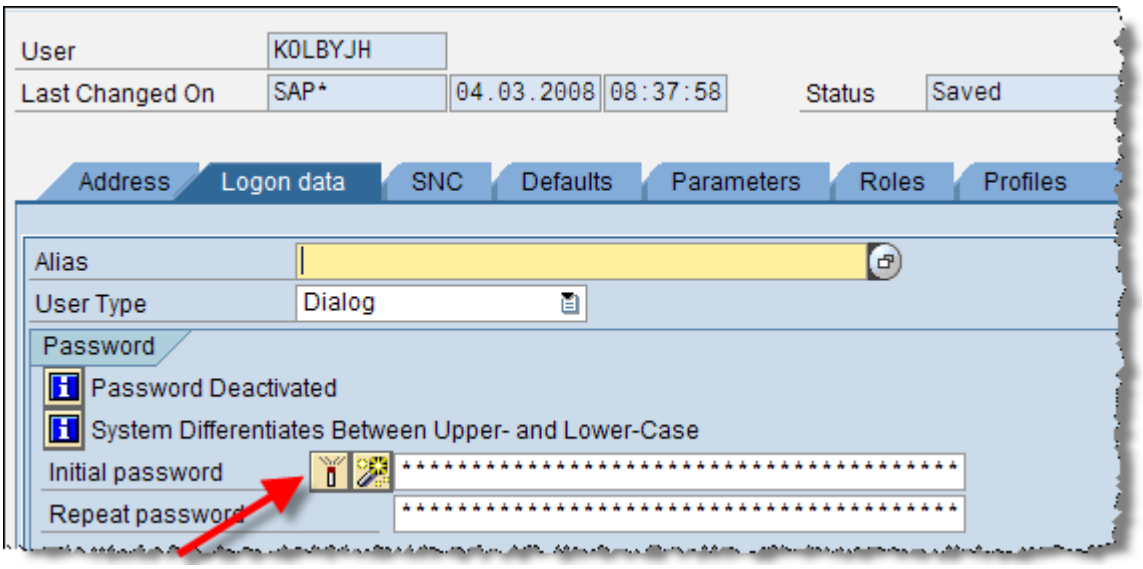

Figure 12. Deactivating the password for a specific user

The process will be the same if SAP's Central User Administration (CUA) is configured.

SAP GUI

The final step is to configure each client workstation. This includes first enabling each Windows workstation for Kerberos and second, configuring SAP system logon entries for single sign-on.

Each workstation will need to be Kerberos-enabled by adding a Windows dynamic-link library (DLL) that implements the GSS API and Kerberos. Fortunately SAP provides a wizard that will automatically restore the DLL on your workstation and set the necessary environment variables. This wizard is attached to SAP note 595341 "Installation issues with Single Sign-On and SNC". Download the file SAPSSO.ZIP and extract it. The ZIP file should contain the file SAPSSO.msi. Copy this file to the Windows client. Sign-on to the Windows client as Administrator or a user with similar rights. Using Windows Explorer, locate the SAPSSO.msi file on the client system and double click the file to start the wizard. At the first dialog screen click "Finish" to complete the installation, see figure 13.

Figure 13. Installing Kerberos support on a Windows workstation

The wizard will automatically install gsskrb5.dll into the \WINDOWS\System32 directory. It will then set the environment variable SNC_LIB=gsskrb5.dll. When this is complete Kerberos should be enabled on your Windows workstation.

The next step is to configure SAP GUI for single sign-on. For this you will need to modify the SAP system logon entry for the specific SAP system that will be accessed via single sign-on. This can be done through SAP Logon or by directly modifying the configuration (INI) file. The following steps describe how to make the changes using SAP Logon.

1. Start SAP Logon
2. Select the SAP system entry to edit and click the “Change Item…” button.
3. Select the “Network” tab
4. Check the box “Activate Secure Network Communication”
5. Once the “Activate Secure Network Communication” box is checked, the SNC name field should be available for input. In this field enter the SNC name which has the form p:<spn>@<kerberos_realm>, see figure 14
6. Select the radio button “Maximum Security Settings Available”
7. Click OK to save

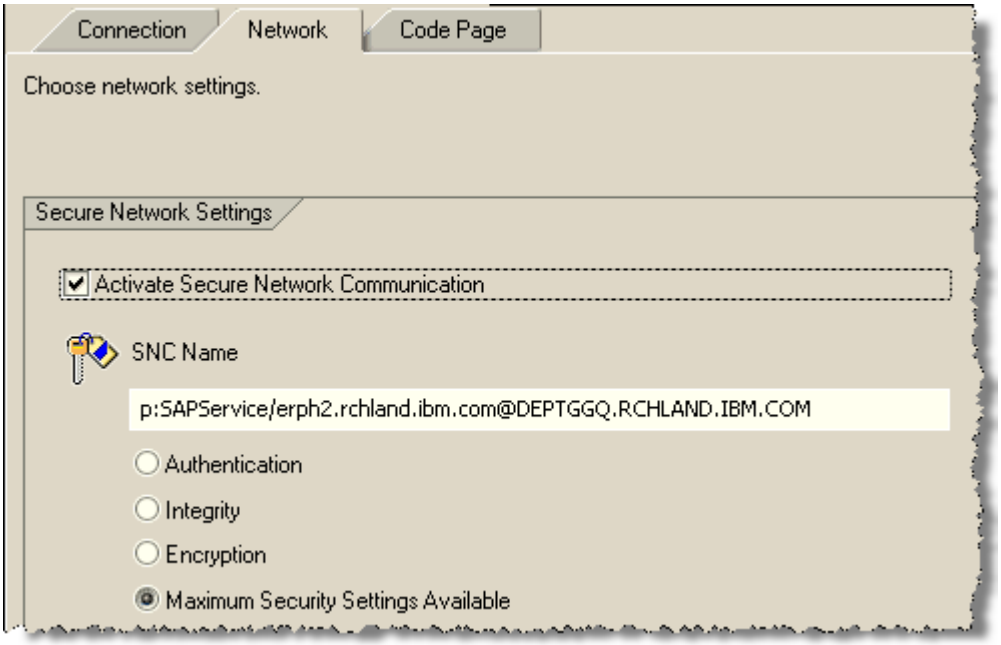

Figure 14. Configuring a SAP system logon entry for single sign-on

Your Windows workstation should now be configured for single sign-on.

Testing the configuration

Finally the moment you have been waiting for. All of the components should now be in place for single sign-on. Make sure that your SAP server has been restarted with SNC enabled. Before attempting to sign-on to your SAP system, make sure that you renew the Kerberos tickets on your Windows workstation. This can easily be done by logging off of the Windows domain and logging back on (you do not need to restart Windows).

After logging on to Windows, you should now be able to sign-on to your SAP application with single sign-on. The process will be the same except you will no longer be prompted for a user and password. If for some reason, your SAP username exists in multiple SAP clients on the same SAP system, a dialog will appear allowing you to choose the appropriate client, see figure 15.

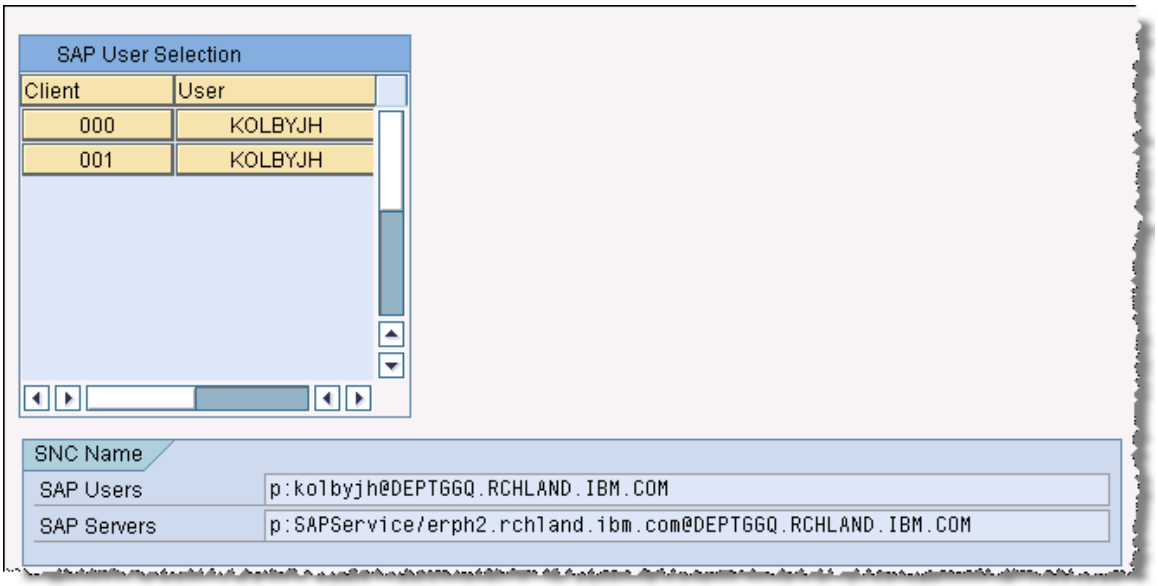

Figure 15. Same username existing in multiple SAP clients

Simply click the user for the client that you wish to use in order to complete the sign-on process. This process will even work for mapping different SAP usernames on the same SAP system to one Active Directory name. For example, you might have a situation where the same person is a business user and an administrator and it is important to keep those roles separate for auditing purposes.

Other Considerations

Once the ABAP Application Server is enabled to use SNC, it expects all incoming communication to also use SNC. This includes internal communications between SAP systems through mechanisms such as RFC or JCO as well as communications from non-SAP systems. Even for Add-in Java Application Server systems, SNC will need to be enabled for the Java system to start, even though the Java and ABAP Application Servers are somewhat integrated. SNC will also need to be enabled for any Java Application Server for which the User Management Engine (UME) repository is located on the ABAP application server.

Enabling the Java Application Server for SNC

Configuring the Java Add-in system is a two step process that involves enabling the Java Application Server for SNC and configuring the system user that communicates with the ABAP Application Server for SNC.

The NetWeaver AS Java ConfigTool is used to make the necessary modifications for SNC. To configure your Java Application Server do the following:

1. Start the ConfigTool
2. Expand Global server configuration
3. Expand services
4. Locate and select the service com.sap.security.core.ume.service see figure 16

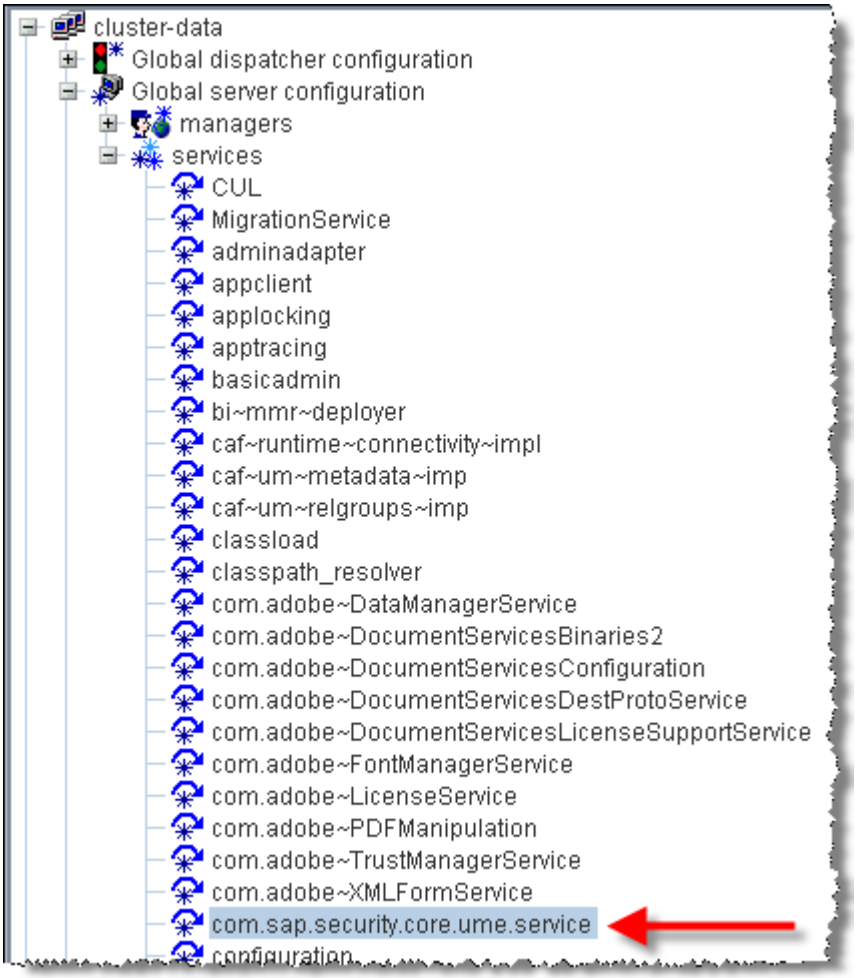

Figure 16. Modifying connection properties

5. A list of global properties will appear in the panel on the right. Locate a set of properties named ume.r3.connection.master.snc.*. These properties will be found towards the bottom of the list. Edit the following properties:

```
ume.r3.connection.master.snc_lib = /lib/libgssapi_krb5.a(libgssapi_krb5.a.so)
ume.r3.connection.master.snc_mode = 1
ume.r3.connection.master.snc_myname = p:<spn>@<kerberos_realm>
ume.r3.connection.master.snc_partnername = p:<spn>@<kerberos_realm>
```

Figure 17 shows the configuration for a Java Add-in system. Since Java Application Server and the ABAP application Server are running on the same host, the parameters for snc_myname and snc_partnername are the same.

ume.r3.connection.master.snc_lib	/lib/libgssapi_krb5.a(libgssapi_krb5.a.so)
ume.r3.connection.master.snc_mode	1
ume.r3.connection.master.snc_myname	p:SAPService/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM
ume.r3.connection.master.snc_partnername	p:SAPService/erph2.rchland.ibm.com@DEPTGGQ.RCHLAND.IBM.COM

Figure 17. Configuring Java Application Server for SNC

Depending on your configuration and whether or not other applications are communicating with the ABAP Application Server, the following instance profile parameters might need to be set:

```
snc/extid_login_diag
snc/extid_login_rfc
snc/accept_insecure_cpic
snc/accept_insecure_r3int_rfc
snc/accept_insecure_rfc
snc/permit_insecure_start
```

More information on the different SNC properties can be found in the SAP Library[8]. Search for the topic “Profile Parameter Settings on AS ABAP”.

Enabling System Users for SNC

By default the Java Application Server connects to the ABAP Application Server using the system user SAPJSF. If SNC is enabled on the ABAP Application Server it is necessary that SAPJSF also be enabled for SNC. The process for configuring a system user for SNC is similar to configuring a regular user for SNC. The only difference is the SNC identity. SAP System users are normally not used by a person to access a system, but by a service or another SAP system. Because SAPJSF will not be used by any person to sign-on to the SAP system, a corresponding Active Directory account will not exist. The SNC identity for SAPJSF will need to be the service principal name used for the SAP system, see figure 18.

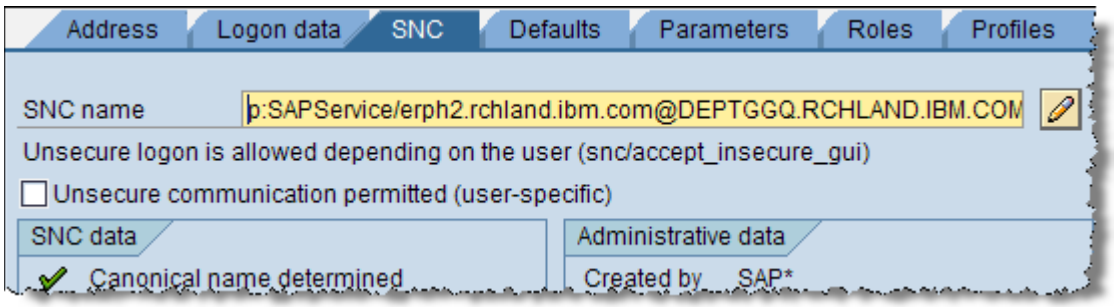

Figure 18. Configure SAPJSF to use SNC

Make sure that the changes made to SAPJSF are done in the client that is specified by ume.r3.connection.master.client. This parameter is part of global server configuration for the service com.sap.security.core.ume.

The Java Application Server will need to be restarted for the changes to take effect.

Conclusion

Now back to our user John. With single sign-on implemented John no longer has any problems managing his usernames and passwords. The only username and password John has to remember is for the Windows domain. When John arrives at work in the

[8] SAP AG. SAP Help Portal. <http://help.sap.com/>.

morning, he logs onto the Windows domain with his workstation. Once logged on, John has the necessary credentials for accessing the other resources in his enterprise. During the course of the day, John normally uses two different SAP systems, each with a different username. With the new single sign-on solution, different SAP usernames can be mapped to the same Windows domain user. So now John doesn't have to remember which profile goes with which system. He simply selects which system he would like to access and through single sign-on the usernames are automatically mapped. John's credentials are passed to the SAP system and he is signed-on. John's user experience and productivity have definitely improved and perhaps more importantly, John never calls the help desk with a password problem.

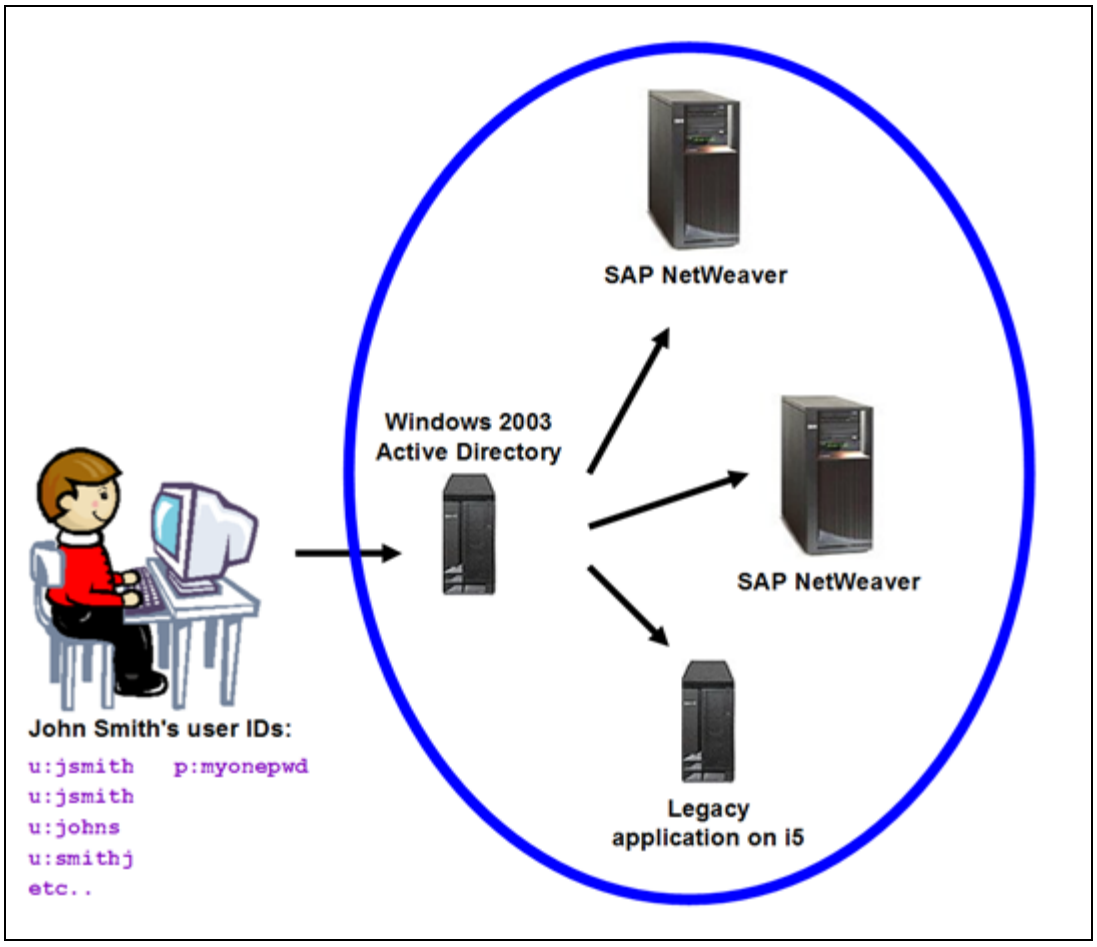

Figure 19. Enterprise environment with a single point of access

As you can see, single sign-on is a solution made up of multiple components, rather than a single product to install and configure. The design and implementation of your single sign-on solution will depend on your environment and design goals. I have shown one approach to expanding single sign-on to the SAP landscape. All of the technology I utilized is either part of Windows, i5/OS, or made available by SAP. No third party technology was necessary, though a number of vendors do offer single sign-on services.

As SAP landscapes grow and become more complex the need for simplifying the management of usernames and passwords at the enterprise level increases. An effective way to simplify and reduce costs is by password elimination through single sign-on. This approach not only improves user experience and productivity, but also decreases administration costs, and improves security.

References

Garman, Jason. *Kerberos: The Definitive Guide*. 1st edition, Sebastopol, CA: Farnham: O'Reilly 2003.

International Business Machines Corporation. IBM System i and i5/OS Information Center: i5/OS PASE. <http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzalf/rzalfintro.htm?resultof=%22%50%41%53%45%22%20/>

International Business Machines Corporation. IBM System i and i5/OS Information Center: Network Authentication Service. <http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzakh/rzakh000.htm?resultof=%22%/>.

International Business Machines Corporation. *System i Security, Single signon, Version 5 Release 4*. Third edition, 2006. <http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzamz/rzamz.pdf/>.

Massachusetts Institute of Technology. Kerberos: The Network Authentication Protocol. <http://web.mit.edu/kerberos/>.

Microsoft Corporation. Microsoft TechNet: Networking and Access Technologies. <http://technet.microsoft.com/en-us/network/default.aspx/>.

SAP AG. SAP Developer Network: Technology – Network Security (BC-SNC). <https://www.sdn.sap.com/irj/sdn/sdnservices/icc?rid=/webcontent/uuid/e112cb72-0501-0010-63a3-f45326c176ae/> .

SAP AG. SAP Help Portal. <http://help.sap.com/>.

SAP AG. SAP Note 595341 "Installation issues with Single Sign-On and SNC".